Chapter 5

Occupancy-based Short-Term Energy Prediction using VMD-LSTM Hybrid Model

Residential and commercial buildings consume significant energy these days [130]. Furthermore, the United States alone is responsible for 40% of global energy use [131], with comparable figures reported by other nations. Maintaining occupant comfort and safety requires a substantial amount of energy. Building energy usage needs to be optimized, and a sufficient number of inhabitants is crucial [132]. To predict short-term energy predictions for a smart building based on occupant counts and other environmental elements like temperature, CO_2, humidity, etc., use a deep learning-based VMD-LSTM. The suggested approach attains 95% accuracy and outperforms the state-of-the-art methods.

5.1 Introduction

Buildings consume vast amounts of energy, and buildings are a significant energy consumption sector today [133]. This is because of Heating, Ventilation, and Air Condition-

ing (HVAC) loads. Nowadays, it is necessary to understand occupant usage patterns and energy consumption analysis to optimize energy consumption in a smart building [134]. Buildings are responsible for greenhouse gases and carbon dioxide emissions because energy demands are higher from buildings than from other sectors. Overall, this accounts for 35% to 40% [135] of total energy consumption. This ratio is even higher in some countries like the United Arab Emirates (UAE), where buildings need more than 70% of the energy produced [136]. Hence, understanding the consumption patterns is crucial to saving energy in buildings [137]. The impact of occupant behavior has become a focus in recent days to reduce buildings' growing energy demand. Various studies on this occupant behavior are investigated to reduce the energy consumption of buildings and reduce the gap between actual and predicted energy consumption [138]. U.S. commercial buildings consume 18% of total energy consumption [51]. Weibull distribution is developed to predict plug load with occupant count. Performs better than linear regression in peak load prediction based on building occupancy [46]. The living standards affect a building's energy consumption and account for 55% of total energy consumption, which is increased yearly by an average of 1.1% [54].

Prediction is a major factor in optimizing energy consumption. To optimize building energy consumption, three types of predicted information are required: occupants in a room, weather in and around the smart buildings, and heat generated from different equipment such as lights, plug loads, etc. [139, 140]. Time is another factor besides occupancy count to predict the energy. Different periods in the form of time, such as occupancy, breaks in intermediate, weekdays, weekends, and vacations, are used to predict energy in a building [141]. Understanding the energy consumption of buildings and the role of humans in designing energy usage is explained. Building activities, energy systems, electrical appliances, and occupants' activities affect the energy consumption of buildings [142]. The detection and estimation of occupancy count in a building also described how occupants' activities help determine buildings' energy con-

sumption. Occupancy measurement errors such as sensor location, camera installation, and measurements affect the performance measurement of buildings' energy consumption. Hence, it is necessary to analyze occupancy patterns and count in a building to predict energy consumption and energy cost savings in the coming days [143]. This work has developed different deep-learning models, such as LSTM, CNN-LSTM, and VMD-LSTM, to predict energy based on occupancy in a smart building.

The major contributions of this chapter are as follows:

- In this work, a novel hybrid deep learning-based VMD-LSTM model has been proposed for forecasting the energy of a building based on the occupancy count.
- The model is analyzed for occupancy count forecasting in the different lag environments, such as minute-wise, 24-hour, or day-wise for smart buildings, and for forecasting energy for different lags in coming days.
- The proposed model has been evaluated and compared with other alternative approaches on error metrics like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R^2 score.

5.2 Proposed Work

Occupancy-based energy forecasting for smart buildings is suggested in this section. Long-Short-Term Memory (LSTM) and Variational Mode Decomposition (VMD) are two components of the suggested approach. The data-driven VMD technique breaks down time series data into Intrinsic Mode Functions (IMFs). VMD is utilized with deep learning methodologies to analyze data and improve the forecasting of complex temporal data. Using optimization, VMD resolves a variational problem. By updating the IMFs until convergence is reached, the VMD recursively minimizes the cost function. The VMD method of signal decomposition is wholly non-recursive. Since the original plug data contains a complex nonlinear connection, using it directly as an input for prediction is exceedingly challenging. Consequently, VMD is utilized to break down

the original plug data into distinct subsequences, which lowers prediction complexity and increases model accuracy. Since every subsequence has a unique set of sparse features, determining the bandwidth and central pulsation is important in the VMD decomposition process.

The Hilbert obtains the unilateral frequency spectrum of every subsequence transform. Next, the hybrid exponential is utilized to find the sub-sequence center frequency and then adjusted to shift it to the bandwidth. Lastly, H' Gaussian smoothing [144] is employed to obtain the bandwidth of each sub-sequence. The constrained results of variational mode decomposition can be expressed as

$$\min_{\{\chi_k\},\{\mu_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\beta_t + \frac{j}{\pi t} \right) * \chi_k(t) \right] e^{-j\mu_k t} \right\|_2^2 \right\} \tag{5.1}$$

Where $\chi_k = \chi_1, \chi_2, \ldots. \chi_k$ is K model component obtained after decomposition. $\mu_k = \mu_1, \mu_2, \ldots. \mu_k$ is central frequency corresponding to each model component. K = Total number of nodes. ∂_t is Partial Derivative. β_t represents impulse function. $*$ is the Convolutional operator. And t is the time script.

Considering both Lagrangian multipliers and quadratic penalty terms, the aforementioned constrained issue is transformed into an unconstrained one. The following is the Lagrangian augmented:

$$L\left(\{\chi_k\}, \{\mu_k\}, \alpha\right) = \gamma \sum_{k=1}^{K} \left\| \partial_t \left[\left(\beta(t) + \frac{j}{\pi t} \right)_2^2 * \chi_t(t) \right] e^{-j\mu_k t} \right\| + \left\| f(t) - \sum_{k=1}^{K} \chi_k(t) \right\|_2^2$$

$$+ \left\langle \alpha(t), f(t) - \sum_{k=1}^{K} \chi_k(t) \right\rangle \tag{5.2}$$

α is Lagrange multiplication operator and γ is quadratic penalty factor. Finally, the iteration-count optimization method is used to solve the original minimization problem, and the updated χ_k and μ_k are as follows:

1) The minimized χ_k as become

$$\hat{\beta_k^{n+1}}(\mu) = \frac{\hat{f}(\mu) - \sum_{i \neq k} \hat{\beta}\mu + \frac{\hat{\alpha(\mu)}}{2}}{1 + 2\gamma(\mu - \mu_k)^2} \tag{5.3}$$

2) The minimized μ_k as become

$$\varphi_k^{n+1} = \frac{\int_0^\infty \varphi \left|\beta_k \hat{\varphi}\right|^2 d\varphi}{\int_0^\infty \left|\beta_k \hat{(\varphi)}\right|^2 d\varphi} \tag{5.4}$$

where n is number of nodes, $\hat{f}(\mu)$, $\hat{\beta}(\mu)$ and $\hat{\alpha}(\mu)$ are the Fourier transform of $f(\mu)$, $\beta(\mu)$ and $\alpha(\mu)$ respectively.

Given that the proposed model combines the VMD and LSTM models, the LSTM is explained as follows. Recurrent Neural Networks (RNNs) perform well on short time series data; nevertheless, they encounter a challenge known as gradient descent when dealing with lengthy time series data. LSTM introduces a cell state into the RNN to solve this. Using input, output, and forget gates in each storage cell, LSTM presents the idea of a multiplicative gate cell. Therefore, with the aid of the introduced cell state, sigmoid activation function, and tanh activation function, long sequence data may be easily predicted and yield superior results.

The LSTM structure consists of an input layer, an output layer, and a hidden layer. The hidden layer consists of an input gate, output gate, forget gate, and cell state, as shown in Figure 5.1. The LSTM consists of an input gate(i_t), forget gate(f_t), cell state(c_t), output gate(o_t), and hidden layer(h_t) to control the flow of information, as shown below.

$$i_t = \sigma(w_{ih} * h_{t-1} + w_{ix} * x_t + b_i) \tag{5.5}$$

$$f_t = \sigma(w_{fh} * h_{t-1} + w_{fx} * x_t + b_f) \tag{5.6}$$

$$c_t = f_t * c_{t-1} + i_t * \varphi(w_{gh} * h_{t-1} + w_{gx} * x_t + b_g) \tag{5.7}$$

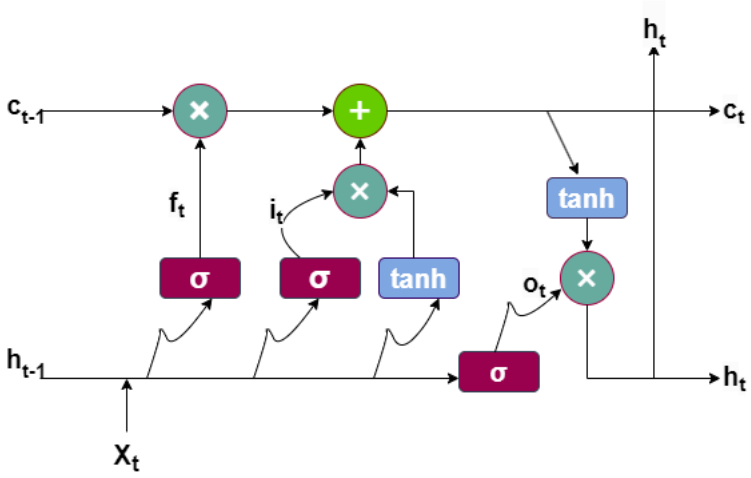


Figure 5.1: Structure of LSTM.

$$o_t = \sigma(w_{oh} * h_{t-1} + w_{ox} * x_t + b_o) \tag{5.8}$$

$$h_t = o_t * \varphi(c_t) \tag{5.9}$$

where w_{ih} and w_{ix}, w_{fh} and w_{fx}, w_{gh} and w_{gx}, and w_{oh}, w_{ox} are weight matrices of input gate, forget gate, cell state, and output gate, respectively. The h_{t-1} is the output of the last hidden layer, the x_t input of the hidden layer, the b_i, b_f, b_g, and b_o are bias terms of input gate, forget gate, cell state, and output gate respectively. The c_{t-1} is the cell state value of the last hidden layer, the σ is the sigmoid function, and the φ is a hyperbolic tangent function (tanh). The proposed work has been demonstrated in Figure 5.2. In this work, the given plug data is decomposed into sub-sequences such as VMD_1, VMD_2, VMD_{n-1},.....VMD_n using VMD; each subsequence data is combined to forecast the electric load using LSTM. Identified different subsequences have different characteristics. Because of this, LSTM with a hyperparameter is used to predict each subsequence separately. Next, all LSTM model results are combined, and a dense layer is used to get the final prediction results. Finally, it uses other machine learning and deep learning models to compare the results obtained from the proposed work, and the proposed work gives better accuracy and outperforms the state-of-the-art.

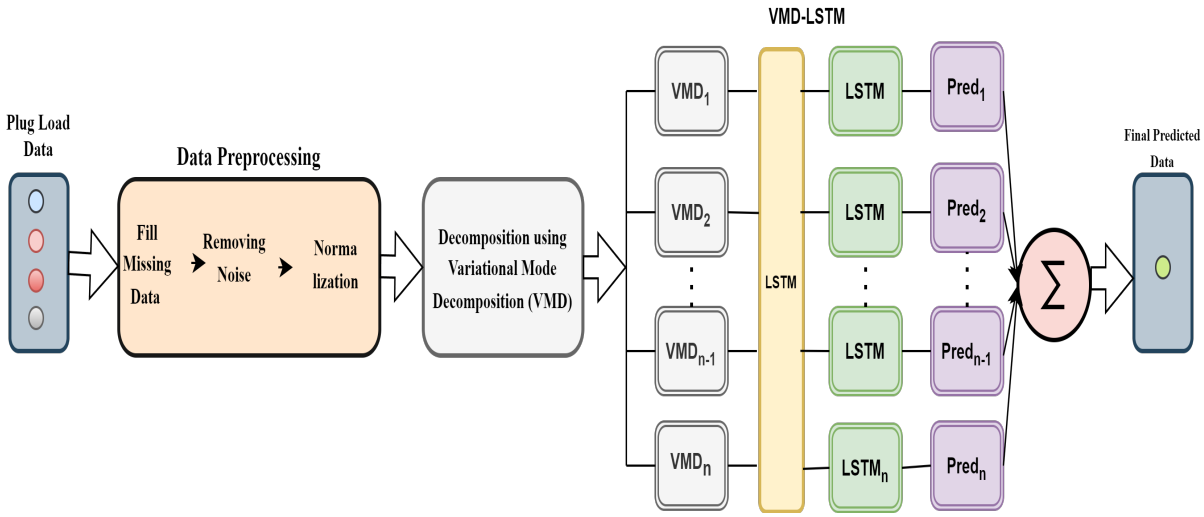


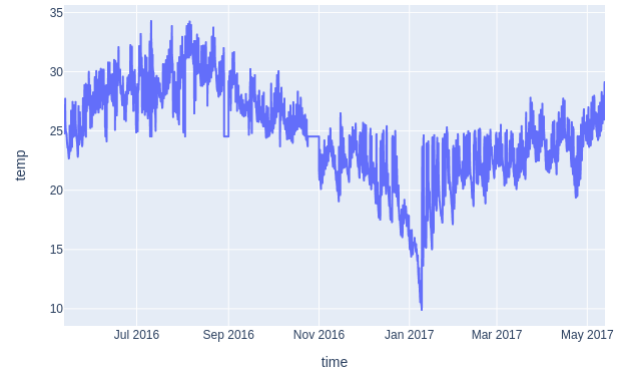
Figure 5.2: Proposed VMD-LSTM for Energy Prediction.

5.3 Experimental Setup

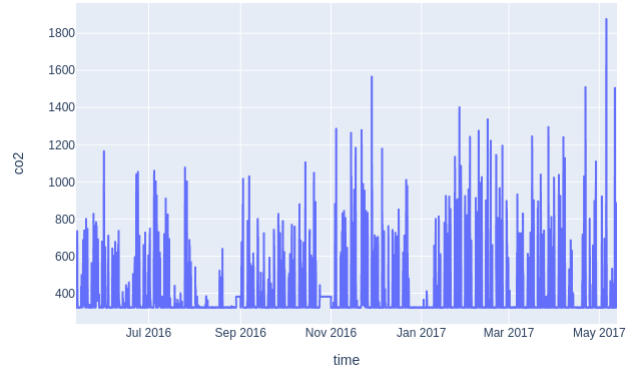
This section comprises the dataset description, data preprocessing, analysis performance indicator, and model comparison. It demonstrates the experimental analysis of the proposed work and takes data from global building occupants to predict energy consumption based on occupancy. The proposed work has been compared with other machine learning and deep learning models such as Linear Regression (LR), Auto Regressive Integrated Moving Average (ARIMA), Long short-term Term Memory(LSTM), and Gated Recurrent Unit(GRU).

5.3.1 Dataset Description

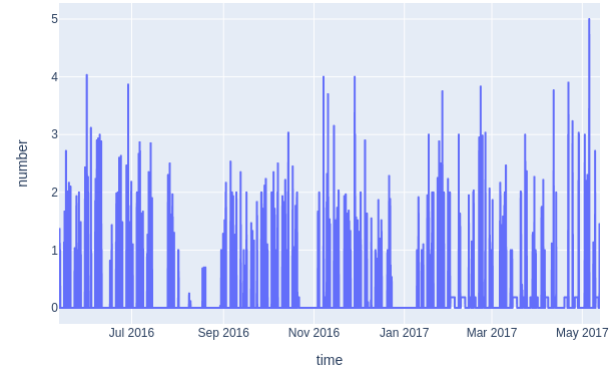
The dataset originates from the global building occupant behavior database maintained by the American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE) and is accessible to the public [145]. Data is gathered from 39 universities spread throughout 10 climate zones worldwide and from commercial and residential structures in 15 nations. The building contains six rooms. Four of the six rooms plug load data are obtained using a variety of sensors, and all six rooms' indoor and outdoor



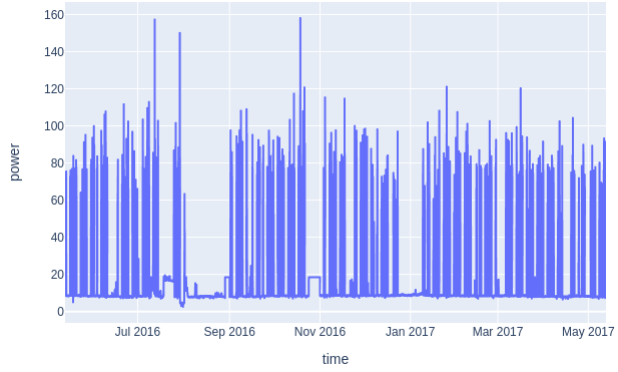
(a) Temperature



(b) CO_2



(c) Occupant_ number



(d) Plug load

Figure 5.3: Raw data of (a) Temperature, (b) CO2, (c) Occupant_ number, and (d) Plug load on energy consumption

temperatures of educational office buildings are recorded. Using camera-based sensors, the occupancy of two rooms recorded for 51 days, from May 22, 2018, to July 11, 2018. The plug and occupancy data contained minute-by-minute information, and additional variables such CO_2, humidity, and temperature are as shown in Figure 5.3

5.3.2 Data Prepossessing

The global occupant behavior database maintained by ASHRAE is the source of the data. The empty columns of the original dataset are removed directly from the raw dataset. Missing values of the original dataset are replaced by -999.. Plug data is subjected to Z-score normalization as it moves from less significant kW values to more significant kW values. The dataset is resampled in several ways, such as 5-minute, 15-minute, 30-minute, and hourly usage. Next, 60%, 20%, and 20% of the total data in the original dataset are allocated to training, validation, and testing, respectively. Data transformation is necessary since data ranges from a few kW to a hundred kW. Therefore, standardize the original data values before transforming the data.

$$\bar{Z}_{newvalue} = \frac{X_t - \mu}{\sigma} \tag{5.10}$$

where $\bar{Z}_{newvalue}$ is new standardized value, μ is mean of sample data and σ is sample standard deviation. The μ and σ is defined as follows

$$\mu = \frac{1}{N}\sum_{t=1}^{N} X_t \tag{5.11}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - \mu)^2} \tag{5.12}$$

Where N is the total number of data samples and X_t is the present value of sample data.

5.3.3 Analysis Performance Indicator

The MAE, RMSE, MAPE, and R^2 are utilized to assess the suggested occupancy-based forecasting model by contrasting it with alternative machine learning and deep learning approaches. The absolute discrepancy between the actual and projected data is the MAE. The average squared difference between the expected and actual data is the RMSE. The average absolute percentage error for each predicted minus actual data point divided by the total amount of actual data is known as the MAPE. The R^2 describes the model's prediction fit. These evaluation metrics are defined as follows.

1. MAE is defined as follows:

$$MAE = \sum_{t=1}^{N} \left| Y_{act}(t) - \hat{Y_{pred}}(t) \right| \tag{5.13}$$

2. RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left(Y_{act}(t) - Y_{pred}(t) \right)^2} \tag{5.14}$$

3. MAPE is defined as follows:

$$MAPE = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{Y_{act}(t) - Y_{pred}(t)}{Y_{act}(t)} \right| * 100 \tag{5.15}$$

4. R^2 is defined as follows:

$$R^2 = 1 - \frac{\sum_{t=1}^{N} \left(Y_{act}(t) - Y_{pred}(t) \right)^2}{\sum_{t=1}^{N} \left(Y_{act}(t) - \overline{Y_{pred}(t)} \right)^2} \tag{5.16}$$

where N is the total number of data points, $Y_{act}(t)$ actual values at time t, $Y_{pred}(t)$ is predicted values at time t and $\overline{Y_{pred}(t)}$ is average of all predicted values at time t.

5.3.4 Models Comparsion

This section describes statistical learning, machine learning, and deep learning models like Linear Regression (LR), ARIMA, LSTM, GRU, and CNN-LSTM to compare the results obtained with proposed occupancy-based energy prediction using VMD-LSTM. The parameters used in the algorithms are expressed in Table 5.1.

Table 5.1: Model names and their parameters

Model Name	**Parameters**
Linear Regression	Input: various lag into different 2 dimensions
ARIMA	The autoregressive terms p=1, difference taken d=1, lagged Previous Error q=2
LSTM	Nodes = 64 , Activation function = Relu , Loss= MAE , Optimizer = Adam.
GRU	Nodes = 64, Activation function = Relu , Loss= MAE, Optimizer = Adam
CNN-LSTM	CNN Filters = 2, Flatten Layer, one layer LSTM, Number of Nodes = 32, Activation function = Relu, Dense, Loss= MSE, Optimizer = Adam, kernel Size =2
VMD-LSTM	alpha = 3, tau = 0, k = 2, DC = 0, Nodes = 64, Dense layer, Loss= MSE, Optimizer = Adam, lookback = 24.

- **Linear Regression (LR):** A linear regression model represents the relationship exit between a dependent variable and the independent variable in a given area or space. The mathematical representation of LR is as follows:

$$y = \delta_0 + \sum_{t=1}^{N} \delta_t x_t + \varepsilon \tag{5.17}$$

where y is dependent variable, x_t is independent variable and $x_t|1 <= t <= N$, ε is residual error and δ is intercept of the response surface with the y-axis and $\delta_t|1 <= t <= N$ (partial regression coefficient)

- **Auto Regressive Integrated Moving Average (ARIMA):**

  The ARIMA statistical model analyses time series data points by determining the difference between the values in the series. It forecasts future data points by taking into account past data points.

The p^{th} order of AR (AR(p)) is represented mathematically as follows:

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + \zeta_t \tag{5.18}$$

and the q^{th} order of and MV (MA(q)) is represented mathematically as follows:

$$y_t = \mu + \zeta_t - \beta_1 \zeta_{t-1} - \beta_2 \zeta_{t-2} - \cdots - \zeta_q \zeta_{t-q} \tag{5.19}$$

And finally, ARIMA *(p, q)* has the following mathematical form of representation:

$$ARIMA(p,q) = \alpha_0 + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \cdots + \alpha_p y_{t-p} + \zeta_t - \beta_1 \zeta_{t-1} - \beta_2 \zeta_{t-2} - \cdots - \zeta_q \zeta_{t-q} \tag{5.20}$$

where y_t is dependent variable at time t, y_{t-1}, y_{t-2}, ... y_{t-p} are dependent variable at different time lags, α_0, α_1 , α_2 ,.....α_p and β_1, β_2,...β_q are estimated coefficient of AR(p) and MV(q), respectively, ζ_t is error term at time t. μ is constant, ζ_{t-1}, ζ_{t-2},.....ζ_{t-q} are error term of previous time periods.

- **Convolution Neural Network (CNN)**: CNN is a deep learning network architecture used mainly for image classification and object recognition, where it may accept two-dimensional data. The CNN can also be used for time series data analysis, where the network accepts one-dimensional data. The CNN performs highly on time series analysis and plugs load energy prediction. The convolution layer is first applied to input data for different attributes or features in the proposed model. Next, it is transferred into feature maps and uses a pooling layer on the acquired feature maps. It is then converted into components to get a more abstract form between the memory cells and hidden layers.
- **Gated Recurrent Unit (GRU)**: The commonly used recurrent neural networks are LSTM and GRU, which are used to overcome the gradient descent problem of

the recurrent network. The gradient descent problems refer to long-term temporal components growing faster than short-term sequences. The LSTM structure consists of cell blocks. In LSTM, the memory block remembers the state through its gates when the cell and hidden states are transmitted from one block to another. The architecture of LSTM and GRU consists of three gates and two gates, respectively. The three gates of LSTM are the input gate, forget gate, and output gate, whereas GRU consists of an update gate and reset gate. The update gate helps understand the memory of the last cell to stay active, and the reset gate helps combine the next cell input sequence with the previous cell memory. The mathematical representation of update gate (Z_t) and reset gate (R_t) are as follows:

$$Z_t = \sigma(W_{zh}h_{t-1} + W_{zh}x_t + b_z) \tag{5.21}$$

$$R_t = \sigma(W_{rh}h_{t-1} + W_{rh}x_t + b_r) \tag{5.22}$$

$$\hat{h_t} = \varphi(W_{hh}r_t + W_{hh}h_{t-1} + W_{hh}x_t + b_h) \tag{5.23}$$

$$h_t = h_{t-1}(1 - Z_t) + Z_t\hat{h}_t \tag{5.24}$$

where W_{zh}, W_{zh}, W_{rh}, and W_{hh} are weight matrices of update, reset, and cell state. x_t is data input, h_{t-1} is cell state vector, h_t is output unit, σ is sigmoid function and φ is hyperbolic tangent function.

5.4 Results and Discussions

An experimental analysis of the proposed VMD-LSTM on occupancy-based energy consumption prediction is presented in this section. This section compares the results obtained from the proposed model with statistical, machine learning, and deep learning models. The results of the LR, ARIMA, LSTM, GRU, and CNN-LSTM models are

compared with the proposed work. VMD is used to decompose IMFs, and it has different characteristics. The LSTM hyperparameter also has different characteristics and helps determine the accuracy prediction. The validity and practicality of the proposed VMD-LSTM are verified and implemented successfully. The analysis of the proposed model is implemented on a computer CPU: Intel Core i7-10700, memory 16GB, and GPU Nvidia GeForce RTX/GTX 3070, Python 3.8 environment using TensorFlow 2.4.0 GPU. The LR has no better values than other deep learning models. The LR's MAE, RMSE, MAPE, and R^2 are 6.37,6.163,12.16, and 0.69, irrespective of the different lags.

5.4.1 Experimental Analysis

The different strategies for energy forecasting for various models and lags in the presence of occupants are examined in this section.

5.4.1.1 5-Minute-Wise Energy Prediction:

The values of the models with various error metrics, including R^2, RMSE, MAE, and MAPE, are displayed in Table 5.2. According to the results, the suggested VMD-LSTM performs better than the ARIMA, LSTM, GRU, and CNN-LSTM models. Regarding 5-minute wise prediction with a 24-hour lag, the suggested model's MAE, RMSE, MAPE, and R^2 are 1.556, 4.493, 7.471, and 96.67%, respectively. As seen in Figure 5.4 (a), the suggested model has low MAE, RMSE, and MAPE compared to other models and high R^2 compared to other models.

5.4.1.2 15 Minute-Wise Energy Prediction:

The energy forecasting error results for a 15-minute prediction using VMD-LSTM with other models are displayed in Table 5.3. The suggested model yields higher R^2 and lower MAE, RMSE, and MAPE than the other models. In the proposed model, MAE, RMSE, MAPE, and R^2 are 2.246, 5.579, 9.130, and 94.52%, respectively. The results

Table 5.2: 5-Minute-Wise prediction of all models.

Error Metric	**Model**	**Lag**				
		2hr	4hr	8hr	12hr	24hr
MAE	ARIMA	8.1936	8.37	8.3917	8.4441	8.4575
	LSTM	2.3765	2.6214	2.4536	2.6251	2.3509
	GRU	2.8325	2.2436	3.0358	2.811	3.2387
	CNN-LSTM	2.5382	2.6382	2.829	3.3257	2.3592
	VMD-LSTM	2.072	1.993	1.5974	1.632	**1.5566**
RMSE	ARIMA	16.3155	16.2826	16.2793	16.7965	16.8168
	LSTM	8.6068	8.5132	8.5796	8.6601	8.5019
	GRU	8.7972	8.4767	8.6746	8.9085	8.8761
	CNN-LSTM	8.5867	8.6217	8.6326	8.8125	8.5746
	VMD-LSTM	7.0819	4.3278	3.8241	4.0008	**4.493**
MAPE	ARIMA	35.403	37.7102	38.0479	35.3138	34.6246
	LSTM	11.6768	14.7498	10.1841	13.1936	10.2184
	GRU	19.6234	9.7527	20.5671	13.1309	20.5236
	CNN-LSTM	17.5225	17.7649	20.9555	32.0453	13.2613
	VMD-LSTM	7.8915	15.6748	9.1821	8.9393	**7.4714**
R^2	ARIMA	0.4717	0.4738	0.4741	0.4401	0.4388
	LSTM	0.8777	0.8804	0.8785	0.8762	0.8807
	GRU	0.8723	0.8814	0.8758	0.869	0.87
	CNN-LSTM	0.8783	0.8773	0.877	0.8718	0.8788
	VMD-LSTM	0.9172	0.9691	0.9759	**0.9736**	**0.9667**

for the related error measures are displayed in Figure 5.4 (b).

Table 5.3: 15-Minute-Wise prediction of all models.

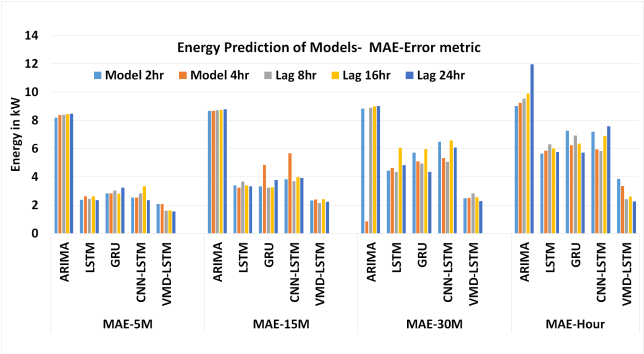
Error Metric	**Model**	**Lag**				
		2hr	4hr	8hr	12hr	24hr
MAE	ARIMA	8.6637	8.6703	8.7111	8.7369	8.7921
	LSTM	3.4019	3.2414	3.6805	3.3965	3.3246
	GRU	3.339	4.8477	3.2321	3.2711	3.7901
	CNN-LSTM	3.8434	5.6616	3.7008	3.9989	3.9131
	VMD-LSTM	2.3269	2.3917	2.1625	2.4153	**2.2467**
RMSE	ARIMA	16.3186	16.6205	16.2872	16.2127	16.2274
	LSTM	9.5058	9.5344	9.6025	9.7791	9.6293
	GRU	9.6873	9.6845	9.6831	9.4955	9.9041
	CNN-LSTM	9.5721	10.5391	9.7367	10.0184	10.0729
	VMD-LSTM	6.5085	6.2491	5.5887	6.2407	**5.5792**
MAPE	ARIMA	42.3238	39.049	42.6784	42.944	44.5878
	LSTM	14.4429	12.8255	16.6316	13.762	13.9347
	GRU	13.4843	26.8606	17.0285	13.5952	15.7677
	CNN-LSTM	29.8344	30.4907	19.0485	17.8549	22.1583
	VMD-LSTM	7.827	10.7298	7.8079	8.9416	**9.1304**
R^2	ARIMA	0.4715	0.45183	0.4736	0.4784	0.4774
	LSTM	0.8404	0.8395	0.8372	0.8314	0.8366
	GRU	0.8342	0.8344	0.8345	0.841	0.8272
	CNN-LSTM	0.8382	0.8039	0.8327	0.823	0.8212
	VMD-LSTM	0.9252	0.931	0.9449	**0.9313**	**0.9452**

5.4.1.3 30 Minute-Wise Energy Prediction:

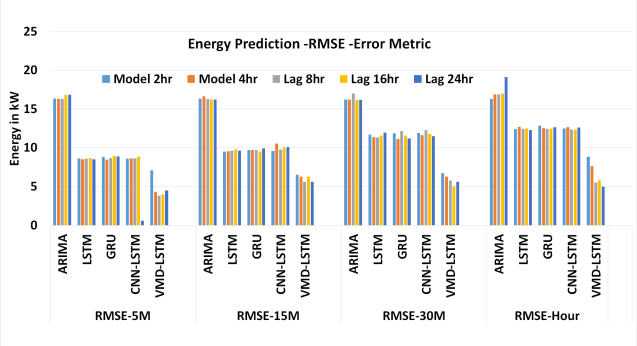
Error metrics for different lags with various models are presented in Table 5.4, corresponding to 30-minute energy prediction. The proposed VMD-LSTM has reduced MAE, RMSE, and MAPE, with error metrics of 2.296, 5.612, and 9.225 as compared to the other state-of-the-art models. The R^2 is higher, at 94.2% than other models. Figure 5.4 (c) displays the error metrics results for various models.

Table 5.4: 30-Minute-Wise prediction of all models.

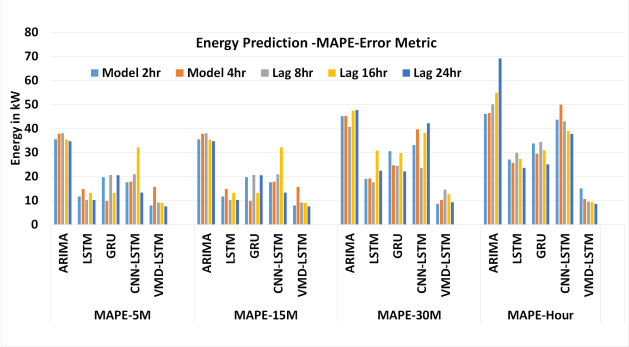
Error Metric	**Model**	**Lag**				
		2hr	4hr	8hr	12hr	24hr
MAE	ARIMA	8.8398	8.8523	8.9082	8.9994	9.0147
	LSTM	4.4451	4.6182	4.3582	6.0697	4.8284
	GRU	5.7154	5.1133	4.9568	5.9605	4.3662
	CNN-LSTM	6.493	5.3429	5.0599	6.5726	6.0891
	VMD-LSTM	2.4901	2.5146	2.8257	2.5541	**2.2963**
RMSE	ARIMA	16.2001	16.2041	17.0164	16.176	16.1758
	LSTM	11.6848	11.3649	11.3295	11.5572	11.9349
	GRU	11.849	11.1278	12.1278	11.5484	11.2046
	CNN-LSTM	11.8797	11.6191	12.2522	11.7852	11.4989
	VMD-LSTM	6.7203	6.2657	5.7644	5.011	**5.6126**
MAPE	ARIMA	45.0655	45.1793	40.703	47.3194	47.6192
	LSTM	18.9559	19.1811	17.6027	30.8251	22.3376
	GRU	30.4762	24.6609	24.391	29.6894	22.144
	CNN-LSTM	33.009	39.557	23.5511	38.1119	42.0921
	VMD-LSTM	8.6417	10.1641	14.4883	12.5455	**9.2252**
R^2	ARIMA	0.4792	0.4789	0.4254	0.4807	0.4807
	LSTM	0.7474	0.7612	0.7629	0.7537	0.7379
	GRU	0.7403	0.7711	0.7283	0.7541	0.769
	CNN-LSTM	0.7389	0.7504	0.7227	0.7439	0.7567
	VMD-LSTM	0.9165	0.9274	0.9386	**0.9537**	**0.942**



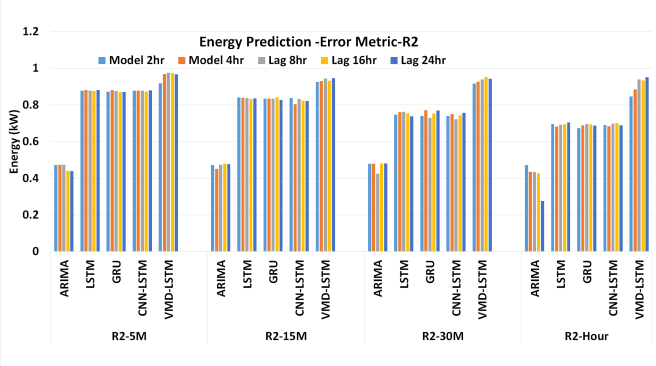
(a) MAE



(b) RMSE



(c) MAPE



(d) R^2

Figure 5.4: Error metrics of (a) MAE (b) RMSE (c) MAPE (d) R^2

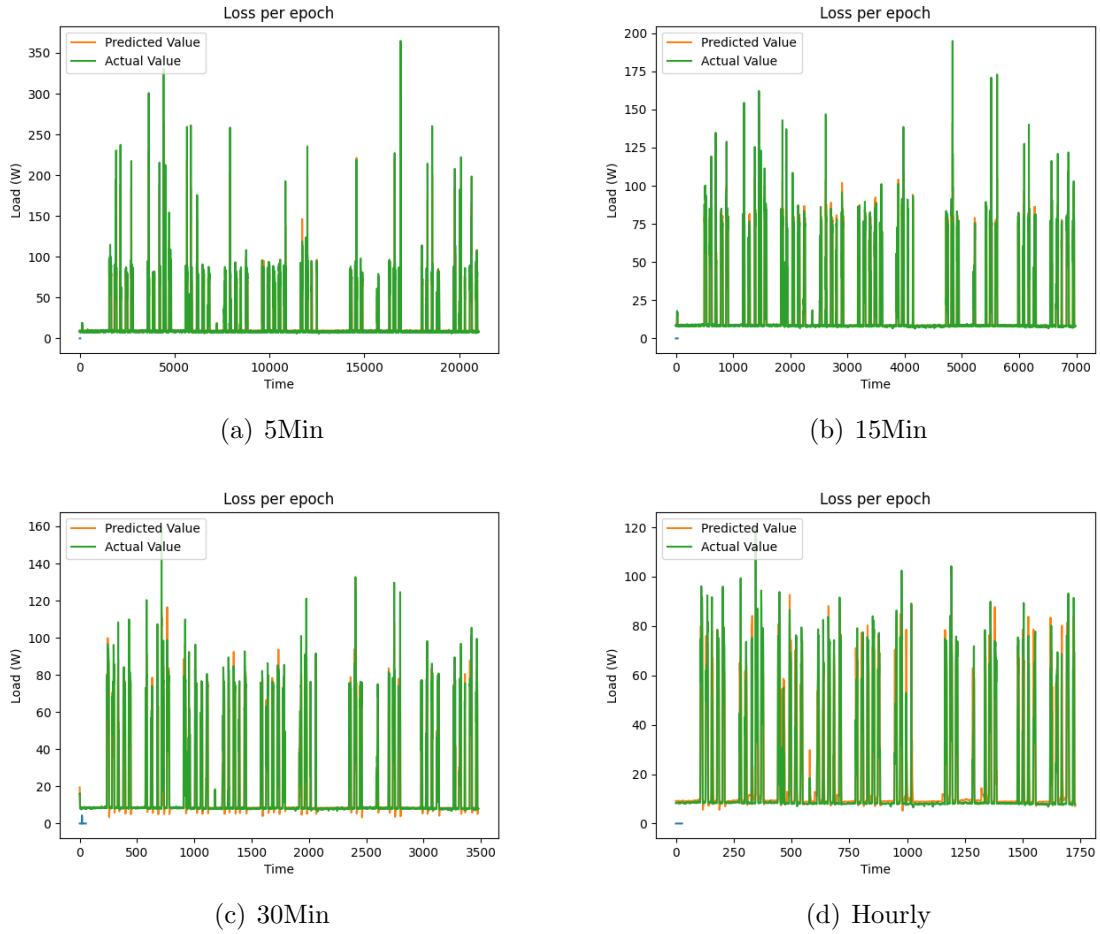
5.4.1.4 Hour-Wise Energy Prediction:

Table 5.5 displays the hourly energy prediction with different models and error metrics. The MAE, RMSE, and MAPE in the proposed model compared to other models are 2.258, 4.985, and 8.581, respectively. It is noted that these values are lower when measured over 24 hours. The model's accuracy is represented by the 95.12% of R^2. In every energy forecasting scenario, the proposed VMD-LSTM outperforms the other models. The related findings with varied delays in hours and all error metrics for the hourly-wise energy prediction are displayed in Figure 5.4 (d).

Table 5.5: Hour-Wise prediction of all models.

Error Metric	**Model**	**Lag**				
		2hr	4hr	8hr	12hr	24hr
MAE	ARIMA	9.0017	9.2449	9.5285	9.9063	11.9575
	LSTM	5.6481	5.8566	6.302	6.0091	5.761
	GRU	7.2538	6.2494	6.931	6.361	5.7235
	CNN-LSTM	7.1863	5.937	5.8282	6.8958	7.5862
	VMD-LSTM	3.8651	3.3447	2.4197	2.6153	**2.2582**
RMSE	ARIMA	16.3112	16.8722	16.89	17.0036	19.1075
	LSTM	12.3788	12.6892	12.4567	12.4656	12.2669
	GRU	12.8495	12.5348	12.4187	12.4335	12.6333
	CNN-LSTM	12.4965	12.6508	12.3661	12.3066	12.6032
	VMD-LSTM	8.8231	7.6205	5.5339	5.7992	**4.9851**
MAPE	ARIMA	46.0112	46.4498	50.076	54.8398	69.0952
	LSTM	27.0334	25.64	29.8137	27.3361	23.5098
	GRU	33.7492	29.3848	34.3152	30.8847	24.9666
	CNN-LSTM	43.5658	49.8668	42.9441	38.9753	37.6993
	VMD-LSTM	14.9804	10.6579	9.4677	9.3511	**8.5811**
R^2	ARIMA	0.472	0.4351	0.4339	0.4262	0.2755
	LSTM	0.6962	0.6811	0.6932	0.6934	0.7047
	GRU	0.6727	0.6888	0.6951	0.6949	0.6868
	CNN-LSTM	0.6904	0.683	0.6976	0.7011	0.6883
	VMD-LSTM	0.8457	0.885	0.9395	**0.9336**	**0.9512**

The results that are presented in Table 5.2 to Table 5.5 indicate that the proposed



(a) 5Min

(b) 15Min

(c) 30Min

(d) Hourly

Figure 5.5: Actual v/s Predicted values using VMD-LSTM for different time horizons (a) 5Min- prediction (b) 15Min- prediction(c) 30Min- prediction (d) Hourly- prediction

VMD-LSTM model performs better than other models with an accuracy of 0.97% and 0.96% for 12-hour and 24-hour predictions in 5 minutes, 0.93% and 0.94% for 12 hour and 24-hour predictions in 15 minutes, 0.95% and 0.94% for 12 hour and 24-hour predictions in 30 minutes, and 0.93% and 0.95% for 12 hour and 24-hour predictions in hour-wise energy prediction for the building. Hence, it is suitable for the prediction of energy for smart buildings in 12-hour lag as well as 24-hour lag.

5.4.2 Forecasting Analysis

This section compares the suggested model's predicting capabilities with various hourly lags. As the proposed VMD-LSTM has better accuracy, it is applied here for energy prediction with occupant count in a building. The actual v/s projected values for the 5-min, 15-min, 30-min, and hour-wise energy prediction for the suggested VMD-LSTM model are displayed in Figure 5.5 (a)-(d), accordingly. Compared to ARIMA, LSTM, GRU, and CNN-LSTM, the prediction of the VMD-LSTM performs better and produces good results in all four categories. The forecasting results for hourly energy projection in a building are shown in Figure 5.5 (a)–(d). Energy prediction is crucial for optimizing building energy use. To do this, three key factors are needed: the number of occupants, weather conditions, and heat from equipment like lights and plug loads. Time also matters, including periods like weekdays, weekends, and breaks. Human behavior, building activities, and appliances all influence energy consumption. Buildings contribute significantly to greenhouse gas emissions, using 35-40% of total energy, and even higher in places like the UAE, where buildings use over 70%. Understanding energy patterns and occupant behavior is key to reducing energy demand and improving predictions. U.S. commercial buildings use % of total energy.

5.4.3 Comparison with existing works

Buildings are responsible for a large portion of greenhouse gases and carbon dioxide emissions because they use more energy than most other sectors. In fact, buildings account for about 35% to 40% of total energy consumption, and in some countries like the UAE, this number goes up to more than 70%. Understanding how energy is used in buildings is important to help reduce consumption. Recently, there has been a focus on how people's behavior in buildings affects energy use, and studies have looked into this to close the gap between predicted and actual energy consumption [135].

In the U.S., commercial buildings use about 18% of total energy. A model called

the Weibull distribution has been developed to predict energy use based on how many people are in a building and performs better than linear regression for predicting peak energy usage [54].

Living standards also play a role in energy consumption, accounting for 55% of total use, and this increases by about 1.1% every year. Predicting energy usage is key to reducing consumption, and for accurate predictions, three main things are needed: the number of people in a room, weather conditions in and around the building, and heat generated by equipment like lights and plug loads. Time is also an important factor in energy predictions, including different periods like weekdays, weekends, and vacation times [142].

5.5 Conclusion and Future work

Using VMD, the occupancy-based plug load data is broken down into various subsequences in this proposed research. The prediction difficulty is decreased by using VMD to separate the initial complex plug data into distinct sub-sequences. Next, to predict the occupancy-based plug load data for other rooms in a building, LSTM is applied to each subsequence. Compared to other machine learning models and deep learning model methodologies, the accuracy result produced after employing VMD-LSTM is superior. The proposed model achieves 95% accuracy for a 5-minute forecast, 95% accuracy for a 15-minute forecast, 94% accuracy for 30-minute forecasts, and 95% accuracy for hour-wise occupancy-based plug load prediction. The proposed model provides more accurate results as VMD separates data into sub-sequences. It can forecast energy for smart buildings with a 12 and 24-hour lag. In future work, occupancy-based load forecasting with deep learning can be combined using edge computing and blockchain to predict future forecasting.

So far, short-term energy forecasting for smart buildings with and without occupancy has been discovered. Still, there is a lot of fluctuation and variation in energy

prediction whenever extreme weather is considered. Typhoons and other weather conditions can cause more noticeable variations in power usage, which makes load forecasting more challenging. Hence, the next chapter proposes short-term energy forecasting based on extreme weather events.